CSS- iSCSI（評価版）

取扱説明書

Cell Storage Service(CSS)を backing store にした iSCSI target

１．０．７

2012/06/11

※最近の disk 利用は、iSCSI によるものが多くなりました。Cell Strage Service のファイルを backing store にした iSCSI target を作成しました。

株式会社トライテック

■改版履歴

版	修正日	作成者	変更点
1.0.1 版	2011/07/27	久保 正業	新規作成
1.0.2 版	2011/08/17	久保 正業	RPM を追加
1.0.3 版	2012/04/10	久保 正業	PSS コンフィギュレーション版
1.0.4 版	2012/04/18	久保 正業	ファイル名のバグフィックス PSS サーバ数変更（5 台→3 台） CentOS6.2 用
1.0.5 版	2012/04/19	久保 正業	CentOS6.2 用 RPM 名のバグフィックス PSS 3 台構成を強調
1.0.6 版	2012/05/18	久保 正業	TGT ベース変更（1.0.18→1.0.27）
1.0.7 版	2012/06/11	久保 正業	バックログの名前を変更（Ubuntu　12.4　LTS 用）

■目次

1. 概要 2
2. 実現方式 3
3. 利用形態例 4
4. 使用方法 6
4.1. マシン構成 6
4.2. パッケージの種類 7
4.3. CSS-iSCSI のインストール 9
4.3.1. CSS サーバ構成の決定と設定ファイルの配布 9
4.3.2. CSS サーバのインストール 9
4.3.3. CSS サーバ制御のインストール 9
4.3.4. iSCSI ターゲットのインストール 9
4.3.5. CSS サーバの初期化と起動 9
4.3.6. CSS サーバ制御の初期化と制御コマンド 9
4.3.7. CSS サーバの終了と再起動 11
4.3.8. CSS サーバのログ 12
4.4. CSS 上にボリューム作成 12
4.5. iSCSI ターゲットの起動と停止 12
4.6. 状態の表示 12
4.7. iSCSI ターゲットのログ 12
4.8. iSCSI イニシエータからの接続（iSCSI　ターゲットの利用） 13
4.8.1. Linux の場合 13
4.8.2. Windows の場合 13

1. 概要

Cell Storage Service (CSS)の高可用性のストレージ機能を使用するひとつのインターフェースとして iSCSI インターフェースの提供形態を用意しました。この iSCSI インターフェースは、以下の特徴があります。

- iSCSI I/F で、CSS の存在を意識せずに、ユーザ選択の自由なファイルシステムで使用できます。
- ディスクとして高価な RAID の冗長性の確保以上のデータの信頼性の向上が図れます。
- サーバを分散すれば、災害にも強く、自然にデータのバックアップを実現できます。

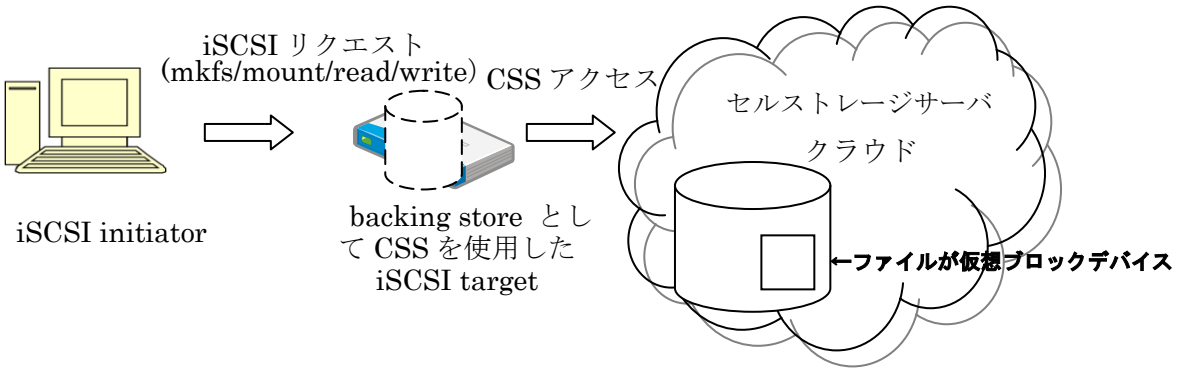


図 1　backing store として CSS を使用した iSCSI target

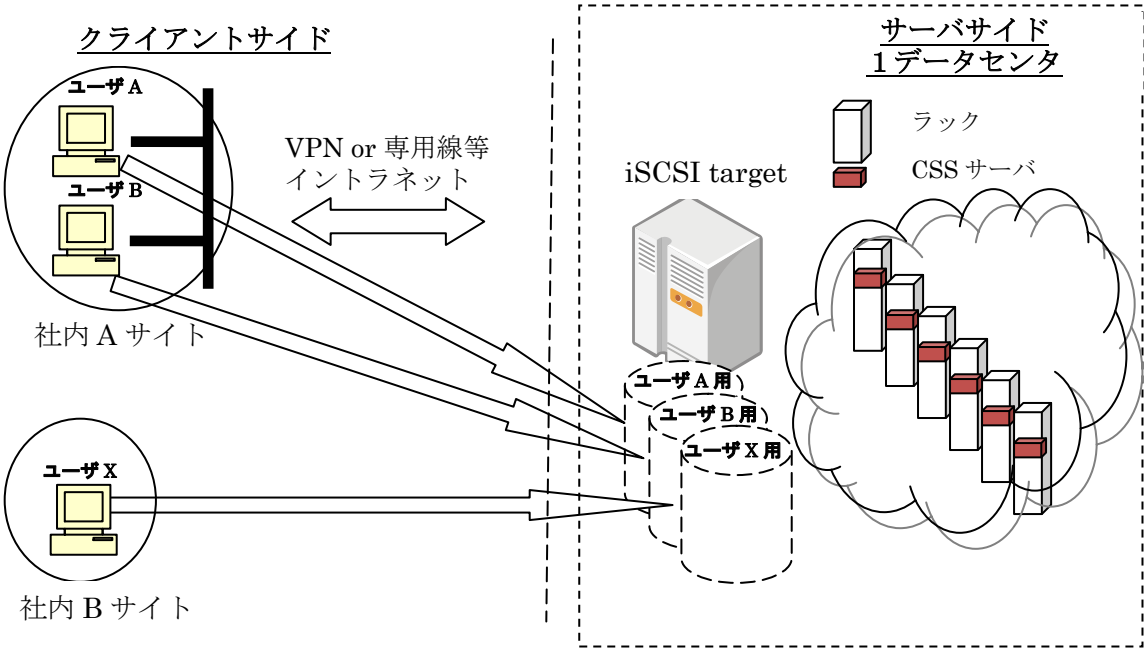
2. 実現方式

提供する iSCSI ターゲットは、TGT version 1.0.27 をもとに backing store のアクセスを通常ファイルから CSS へのアクセスに変更したものです。「４．使用方法」にしたがって、インストール/環境設定/デーモン起動/CSS タイプの LUN 作成していただければ、ネットワークに公開可能です。

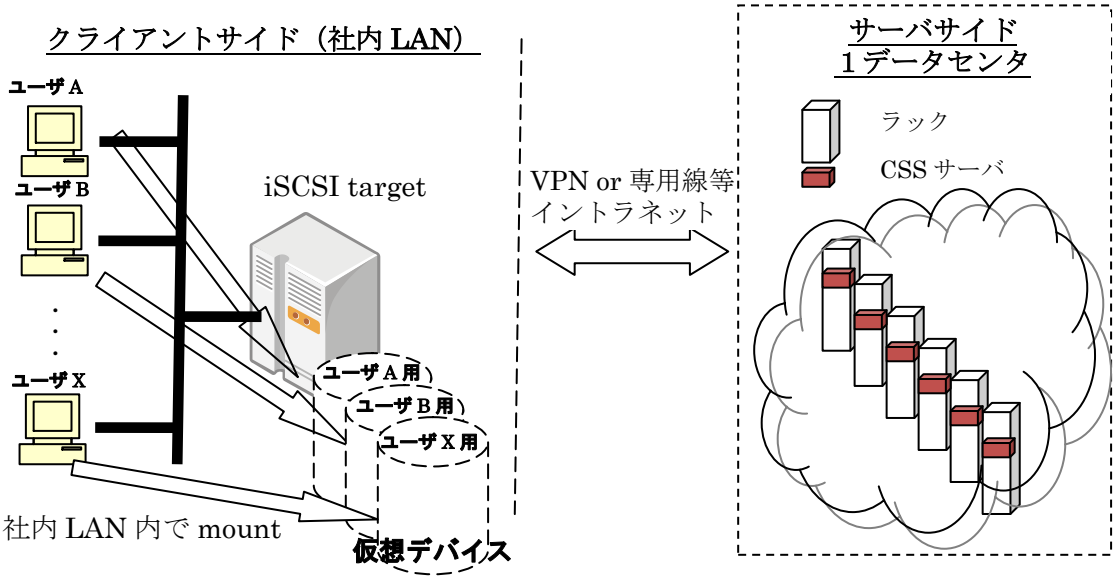
CSS（Cell Storage Service）のほうは、iSCSI　target のデーモン起動前に（正確には、LU(ロジカルユニット)作成前に）デーモンプログラムからアクセスできるようにサーバを配置する必要があります。その方法は、「４．使用方法」に詳しく記述してあります。

3.　利用形態例

（1） iSCSI target と CSS サーバをサーバサイドに置いて、場所が分かれていても社内の別の場所からでもアクセスできます。

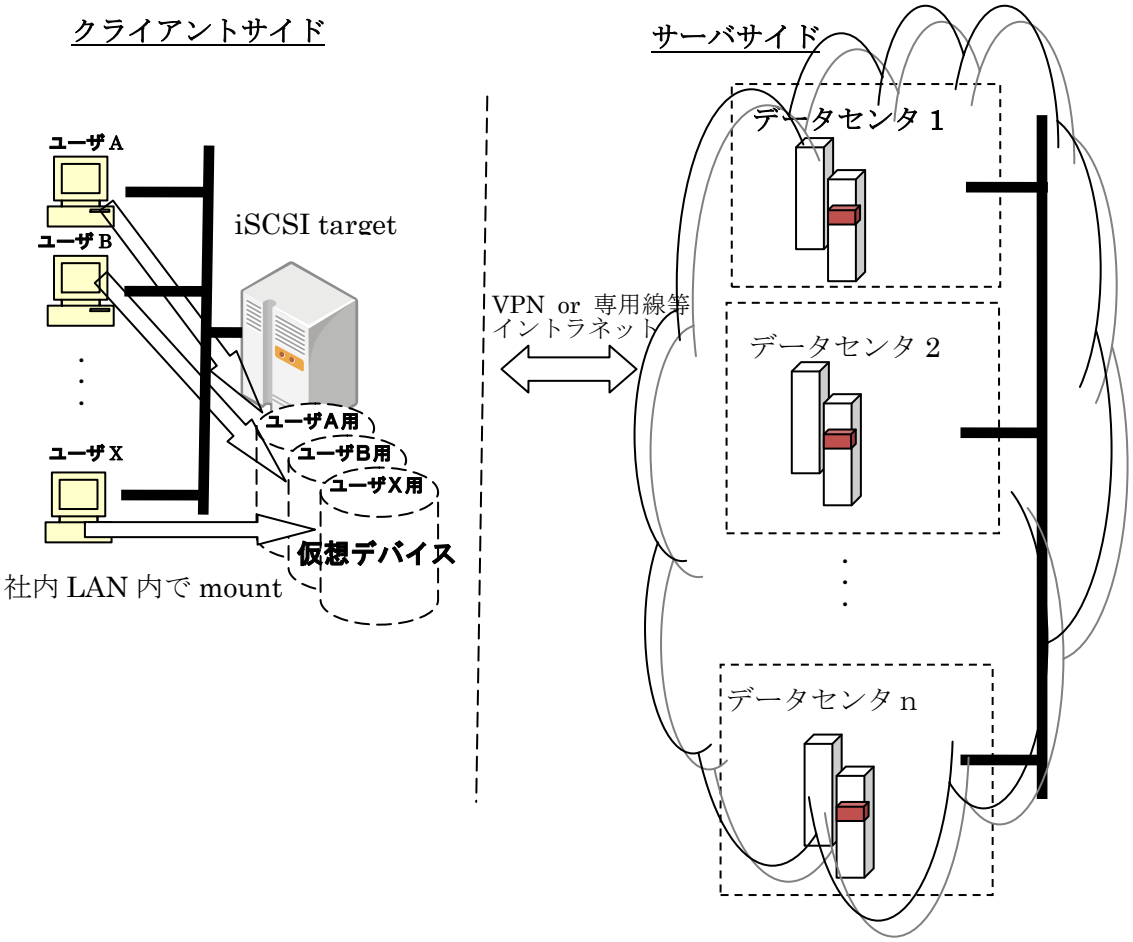


（2） クライアントサイドに iSCSI target を置いて CSS サーバをデータセンタに置いて、データと処理を分散する。極端には,initiator/target 同マシンの構成もできます。



（３） CSS サーバ分散配置

複数のデータセンタを使って、CSS サーバを分散。危機管理を強化する。



4. 使用方法

4.1. マシン構成

（１） CSS サーバマシン（VM でも可）2 台以上 4 台まで

（２） iSCSI ターゲット（VM でも可）1 台（※CSS サーバ制御マシンと同一でもよい）

（３） CSS サーバ制御（VM でも可）1 台（※iSCSI ターゲットマシンと同一でもよい）

OS は、Ubuntu12.4(amd64) LTS server 版か CentOS6.2(Final)でお願いします。

それぞれのマシンには、共通のユーザと SSH（クライアントとサーバ）をいれてください。

また CentOS で実行する場合は、ファイアウォールの設定に気を付けてください。CSS サーバは、使用するポート（例題では、50010～50013）をそれぞれ対応するマシン上で通信を許可し、iSCSI ターゲットマシンでは、3260 ポートの通信を許可しておいてください。たとえば、RH-Firewall-INPUT chain にファイアウォールの設定がある場合は、以下のようなフィルタをその「RH-Firewall chain」に追加しておいてください。

udp のフィルタ追加

-A RH-Firwall-1-INPUT -p udp -m udp -j ACCEPT ←これは、dport 規制を外す意味です。

tcp のフィルタ追加

-A RH-Firwall-1-INPUT -p tcp -m tcp --dport 50010:50013 -j ACCEPT ←各 CSS サーバ 1 ポートでも可。

-A RH-Firwall-1-INPUT -p tcp -m tcp --dport 3260 -j ACCEPT ←iSCSI ターゲットの場合。

また、（２）iSCSI ターゲットマシンには、libconfig-general-perl（Ubuntu）、perl-Config-General（CentOS）パッケージをインストールしておいてください。

以下に、マシン構成の例を示します。

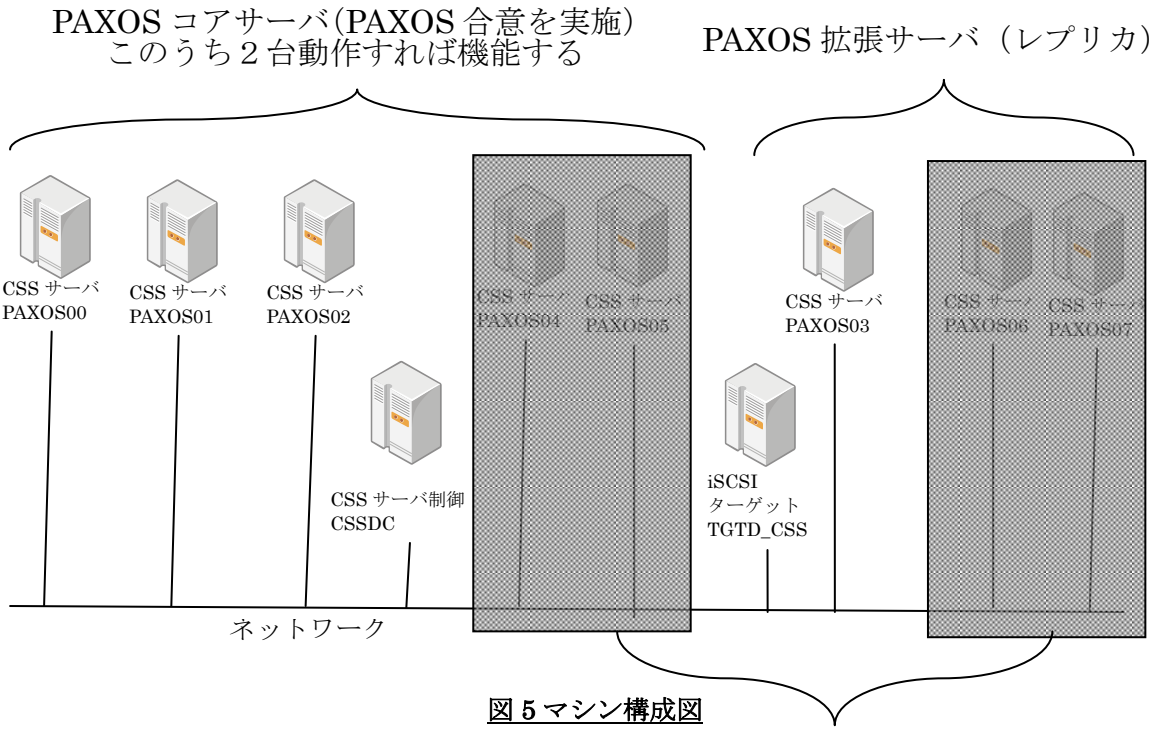


図 5 マシン構成図

4.2. パッケージの種類

以下のパッケージと設定ファイル（例）があります。

（１） CSS 共通

ファイル名：csscom.deb(ubuntu),css-common-1.1.1-1.el6.x86_64.rpm(CentOS)

パッケージ名：css-server

（２） CSS サーバ

ファイル名：cssd.deb(ubuntu),css-server-1.1.1-1.el6.x86_64.rpm(CentOS)

パッケージ名：css-server

（３） CSS サーバ制御

ファイル名：cssdc.deb(ubuntu),css-server-controller1.1.1-1.el6.x86_64.rpm(CentOS)

パッケージ名：css-server-controller

（４） iSCSI ターゲット

ファイル名：tgtd.css.deb(ubuntu),tgt-css-target-1.1.1-1.el6.x86_64.rpm(CentOS)

パッケージ名：tgt-css-target

（５） server.conf

/etc/css ディレクトリにおいて、CSS サーバ構成を記述したものです。この設定ファイルは、CSS サーバマシン、CSS サーバ制御マシン、iSCSI ターゲットマシンの全てに必要です。最初に CSS サーバの構成を決めた時に全マシンは配布

しておいてください。図 5 と同じ CSS サーバ構成で、サーバの DNS(または /etc/hosts での指定した)名を paxos00~paxos03 としたとき以下のようになります。

```
# id server udp_port tcp_port path
0 paxos00 50000 50010 /css
1 paxos01 50001 50011 /css
2 paxos02 50002 50012 /css
3 paxos03 50003 50013 /css
```

1 列目の id のところには、0~3 を書きます。サーバ番号で役割は、0~2 まではコアサーバといって、実際の PAXOS 合意を行うサーバで、どれも平等で役割の違いはありません。3 は、PAXOS 合意には参加しませんが、リプリケーションのように 0~2 までのコアサーバと同じデータを保持します。

2 列目の server のところは、サーバ名または IP アドレスを記述します。

3,4 列目は、サーバ同士やサーバクライアント間で使用する UDP/TCP ポート番号です。同一ホストで動作する他のプログラムとポート番号と重ならないように設定してください。

5 列目の path のところは、データの保持場所で iSCSI で見えるボリューム（複数可能）の置き場所（制御データおよび内部の一時データも含む）です。確保したサイズだけのデータを作成できます。複数パス(デバイス)は使用できません。

※ *デバイスは指定できません。必ずフォーマットしてマウントしておいてください。*

最低で 0~2 のうちの 2 台があれば CSS は動作しますが、server.conf は、4 レコード必要で、2 列目の server のところは、DNS 又は/etc/hosts で名前解決ができるか、IP アドレスでなければなりません。

（６） targets.conf

iSCSI ターゲットマシンの/etc/tgt に置いて、ターゲット名、対応する CSS ファイル名、アクセス可能ユーザ名/パスワード等を指定します。例では、以下のようなものです。

```
<target iqn.2001-03.jp.tritech:tgtd-css>
    <backing-store vol001>
        bs-type css
        lun 1
    </backing-store>
    incominguser iscsi_css iscsi_css000
</target>
```

target タグにターゲット名（IQN）を指定します。複数のターゲットを作成する場合は、<target>~</target>を繰り返します。target タグ中の backing-store タグで、CSS ファイル名を指定します。このバッキングストアは、createISCSIVol コマンドで作成します（後述）。backing-store タグ中に bs-type と lun を指定します。bs-type は、css を必ず指定します。これによって、CSS-iSCSI を使用することになります。lun は、論理ユニット番号で 1 つのターゲッ

トで複数のバッキングストアを使用する場合に区別するための番号です。ターゲットでユニークにした 1 以上の番号をバッキングストアに与えます。

incominguser は、そのターゲットにアクセスできるユーザをユーザ名とパスワードを指定して制限するものです。

※ initiator-address を指定してログイン可能なイニシエータの制限もできます。

4.3. CSS-iSCSI のインストール

4.3.1. CSS サーバ構成の決定と設定ファイルの配布

マシン構成を決めます。たとえば、CSS サーバの DNS 名を paxos00~paxos04 をとし、CSS サーバ制御マシンの DNS 名を cssdc、iSCSI ターゲットマシンの DNS 名を tgtd_css とした場合に、CSS サーバ構成（CSS サーバのデータパスを含めて）を server.conf に記述して(4.2 パッケージの種類（5）参照)、全マシンに配布して/etc/css に置きます。

4.3.2. CSS サーバのインストール

paxos00~paxos03 に CSS サーバパッケージをインストールします。

sudo dpkg -i csscom.deb cssd.deb(ubuntu)

rpm -ivh css-common-1.1.1-1.el6.x86_64.rpm css-server-1.1.1-1.el6.x86_64.rpm(CentOS)

※ CentOS の場合は、root で実行してください。以下同様です。

4.3.3. CSS サーバ制御のインストール

cssdc に CSS サーバ制御パッケージをインストールします。

sudo dpkg -i csscom.deb cssdc.deb(ubuntu)

rpm -ivh css-common-1.1.1-1.el6.x86_64.rpm css-server-controller-1.1.1-1.el6.x86_64.rpm (CentOS)

4.3.4. iSCSI ターゲットのインストール

tgtd_css に iSCSI ターゲットパッケージをインストールします。

sudo dpkg -i csscom.deb tgtd.css.deb(ubuntu)

rpm -ivh css-common-1.1.1-1.el6.x86_64.rpm tgt-css-target-1.1.1-1.el6.x86_64.rpm (CentOS)

4.3.5. CSS サーバの初期化と起動

paxos00~paxos03 で、初期化してサーバをスタートします。

Ubuntu の場合:

sudo css_delper.sh css

sudo /etc/init.d/cssd start

CentOS の場合(root で行う) :

css_delper.sh css

/etc/init.d/cssd start

4.3.6. CSS サーバ制御の初期化と制御コマンド

cssdc で、CSS サーバへのパスワードなしログインを可能にするため ssh のパスワードなしキーを作成します。

ssh-keygen

Enter file in which to save the key (/home/kubo/.ssh/id_rsa):

とでたら、入力なしでEnterを入力

Enter passphrase (empty for no passphrase):

とでたら、入力なしでEnterを入力

Enter same passphrase again:

とでたら、入力なしでEnterを入力

これで、~/.ssh/id_rsa.pubができます。そして、id_ras.pubをpaxos00~paxos03に配布して~/.ssh/authorized_keysに設定（ホストcssdcでcss_set_ssh.sh cssを実行してください)。

これで、各コマンドがスムーズに実行可能になります。以下は、ホストcssdcで実行可能です。

（1） サーバ生死確認

css_lookup-server.sh セル名

以下のように表示（0~3 にないものがあれば死んでいます）

9326 ? Sl 2:26 /css/PFSServer css 0 /css/PFS-0

13005 ? Sl 3:47 /css/PFSServer css 1 /css/PFS-1

5133 ? Sl 2:10 /css/PFSServer css 3 /css/PFS-3

（2） CSSのファイル操作コマンド

PFSClient css readdir / ← パス名指定

PFSClient css stat /vol001 ← ファイル状態取得

PFSClient css read /vol001/00/00/01 file1 ← ファイル取得

etc. …….

（3） CSSサーバシャットダウン

SessionShutdown css 0 ← id

（4） CSSサーバ状態取得

PFSProbe css 0 accept ← 接続クライアント情報表示

PFSProbe css 1 outbound ← 保持中のアウトバウンドデータ

PFSProbe css 2 fd ← オープン中のファイル情報表示

etc….

（5） Paxos情報取得

PaxosAdmin css 0 control ← サーバ間の接続情報

PaxosAdmin css 1 paxos [ページ] ← PAXOS合意の表示

etc….

（６）　ポート（tcp/udp）情報取得

サーバプロセスが突然ダウンした場合に、すぐ立ち上がらないことがあります。固定ポートをバインドできない場合です。そのために各サーバのポート情報を表示するコマンドを用意しました。

css_lookup_port.sh css [id] ← idを指定するとそのidのcssサーバの情報

以下のように表示されます。

```
tcp        0      0 10.101.1.211:4010       0.0.0.0:*               LISTEN
tcp        0      0 10.101.1.211:4010       10.101.1.80:33336       ESTABLISHED
tcp        0      0 10.101.1.211:4010       10.101.1.201:58496      ESTABLISHED
・・・・・・・
・・・・・・・
udp        0      0 10.101.1.211:4000       0.0.0.0:*
・・・・・・・
```

4.3.7. CSS サーバの終了と再起動

CSS サーバを終了させるためには、

sudo /etc/init.d/cssd stop

です。

初期起動以外のサーバの起動は、3 つの場合にわけられます。

１）　最初の 1 台目のサーバの立上

全てのサーバが落ちてしまった（あるいは落とした）場合に、立上げ手順が必要になり、まず信頼できるデータを持ったサーバを id が 0~2 までのコアサーバのうちから 1 つ選んで最初に立上げます。

sudo /etc/init.d/cssd start

２）　2 台目までのコアサーバの立上

コアサーバ（id が 0~2 までのサーバ）が 2 台立上がってない（1 台立ち上がっている）場合は、データを参照するサーバ ID を指定して立上げます。（指定サーバのデータを参照してそれに合わせてから動作する）

以下のような立上になります。

sudo /etc/init.d/cssd restart autonomic id ← 「autonomic」は、そのまま文字列”autonomic”と入力。id は、先に立ちあがっているデータを参照するサーバの ID

３）　コアサーバ過半数（2 台以上）立ち上がっている場合

コアサーバ過半数立上っている場合は、データを参照するサーバ ID を指定せず、PAXOS 合意で決まったマスタサーバ（指定なし）を参照して立

上げできます。

sudo /etc/init.d/cssd restart autonomic

4.3.8. CSS サーバのログ

CSS サーバのログは、シスログ（/var/log/syslog(ubuntu),/var/log/messages(CentOS)）にでます。タグは css_0 等 css_サーバ ID です。ログである程度の状況を把握できます。

※ logger コマンドを使用していますので、user.notice レベルのログを出します。シスログの設定でログを出す場所を変更できます。

4.4. CSS 上にボリューム作成

CSS 上に作成するターゲットが使用するボリューム（server.conf で指定した path とは違います）を作成します。任意の名前でよいですが、targets.conf に書いた名前のボリューム名（バッキングストア名）のボリュームを作成します。

createISCSIVol css volName num{G|T}

ex. createISCSIVol css vol001 1T ← 1T バイトのボリュームをボリューム名「vol001」として作成します。

※ createISCSIVol コマンドは、iSCSI ターゲットをインストールしたホストの/sbin にインストールされます。

4.5. iSCSI ターゲットの起動と停止

起動： sudo /etc/inid.d/tgtd.css start(ubuntu)

/etc/inid.d/tgtd.css start(CentOS root で実施)

停止： sudo /etc/inid.d/tgtd.css stop

/etc/inid.d/tgtd.css stop(CentOS root で実施)

4.6. 状態の表示

iSCSI ターゲットの状態やアカウントの状態は以下のようにして表示します。

Ubuntu の場合：

sudo tgtadm --lld iscsi --op show --mode target ←iSCSI target

（あるいは、 sudo tgt-admin –s）

sudo tgtadm --lld iscsi --op show --mode accunt ←アカウント

CentOS の場合(root で実施)：

tgtadm --lld iscsi --op show --mode target

（あるいは、 tgt-admin –s）

tgtadm --lld iscsi --op show --mode accunt

4.7. iSCSI ターゲットのログ

CSS サーバのログは、シスログ（/var/log/syslog(ubuntu),/var/log/messages(CentOS)）にでます。タグは tgtd です。ログである程度の状況を把握できます。

※　daemon.error と daemon.debug レベルのログを出します。シスログの設定でログを出す場所を変更できます。

4.8. iSCSI イニシエータからの接続（iSCSI　ターゲットの利用）

実際のクライアント（iSCSI initiator）は、どのような手続きで Paxos iSCSI を利用できるのかを以下に説明します。

4.8.1. Linux の場合

Linux の場合は、以下の手順で iSCSI ターゲットに接続します。以下全て root 権限で行います。

（1） ユーザ/パスワードの設定

/etc/iscsi/iscsid.conf の node.session.auth.username と node.session.auth.password に接続する iSCSI ターゲットに設定したアカウントを書きます。

node.session.auth.username = iscsi_css

node.session.auth.password = iscsi_css000

（2） iSCSI initiator の起動(open iscsi)

/etc/init.d/open_iscsi start

（3） iSCSI target の探索

iscsiadm --mode=discovery --type=sendtargets　-p dhcp1140

以下出力例（iSCSI ターゲット名が列挙されます）です。

[dhcp1140]:3260,1 iqn.2001-03.jp.tritech:dhcp1140-css

[dhcp1140]:3260,1 iqn.2001-03.jp.tritech:dhcp1140-css-shojima

[dhcp1140]:3260,1 iqn.2001-03.jp.tritech:dhcp1140-css-kubo

[dhcp1140]:3260,1 iqn.2001-03.jp.tritech:dhcp1140-css-nw

[dhcp1140]:3260,1 iqn.2001-03.jp.tritech:dhcp1140-css-nami

※ dhcp1140 は、iSCSI ターゲットの動作しているホスト名（IP アドレスでも可）

（4） 接続したい iSCSI ターゲットへのログイン

iscsiadm --mode=node --login --targetname iqn.2001-03.jp.tritech:dhcp1140-css

これで新しいデバイスとして,/dev/sd*が見えます。

（5） フォーマット

最初は未フォーマットなので、必要なら fdisk 等でパーティションを切って、mkfs でファイルシステムを作成します。

（6） マウント

Linux のファイルシステムとしアクセスするために、mount コマンドでどこかのパスにマウントして使用します。

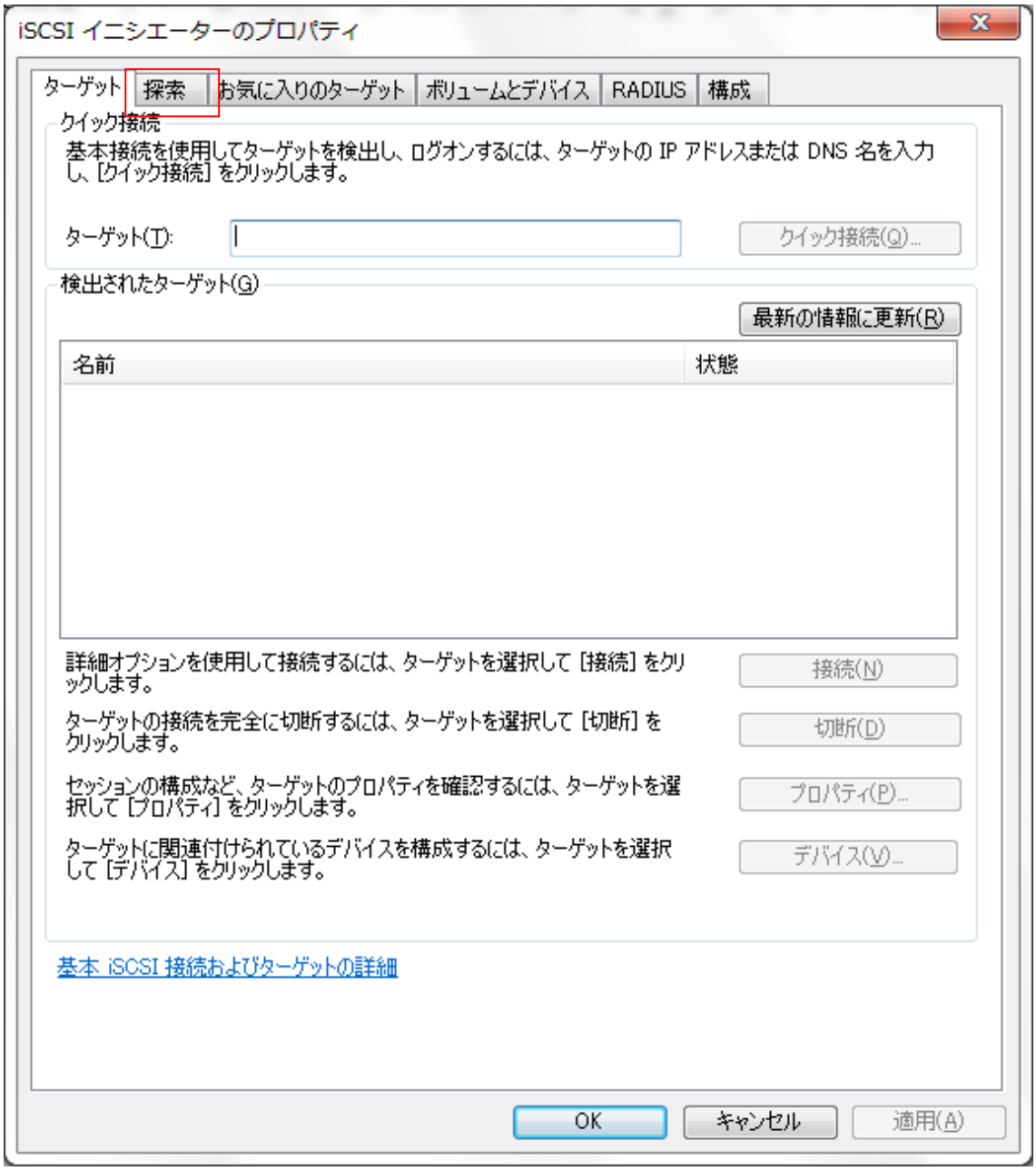
4.8.2. Windows の場合

Windows の場合は、Windows7 の場合について以下の手順を例示します。

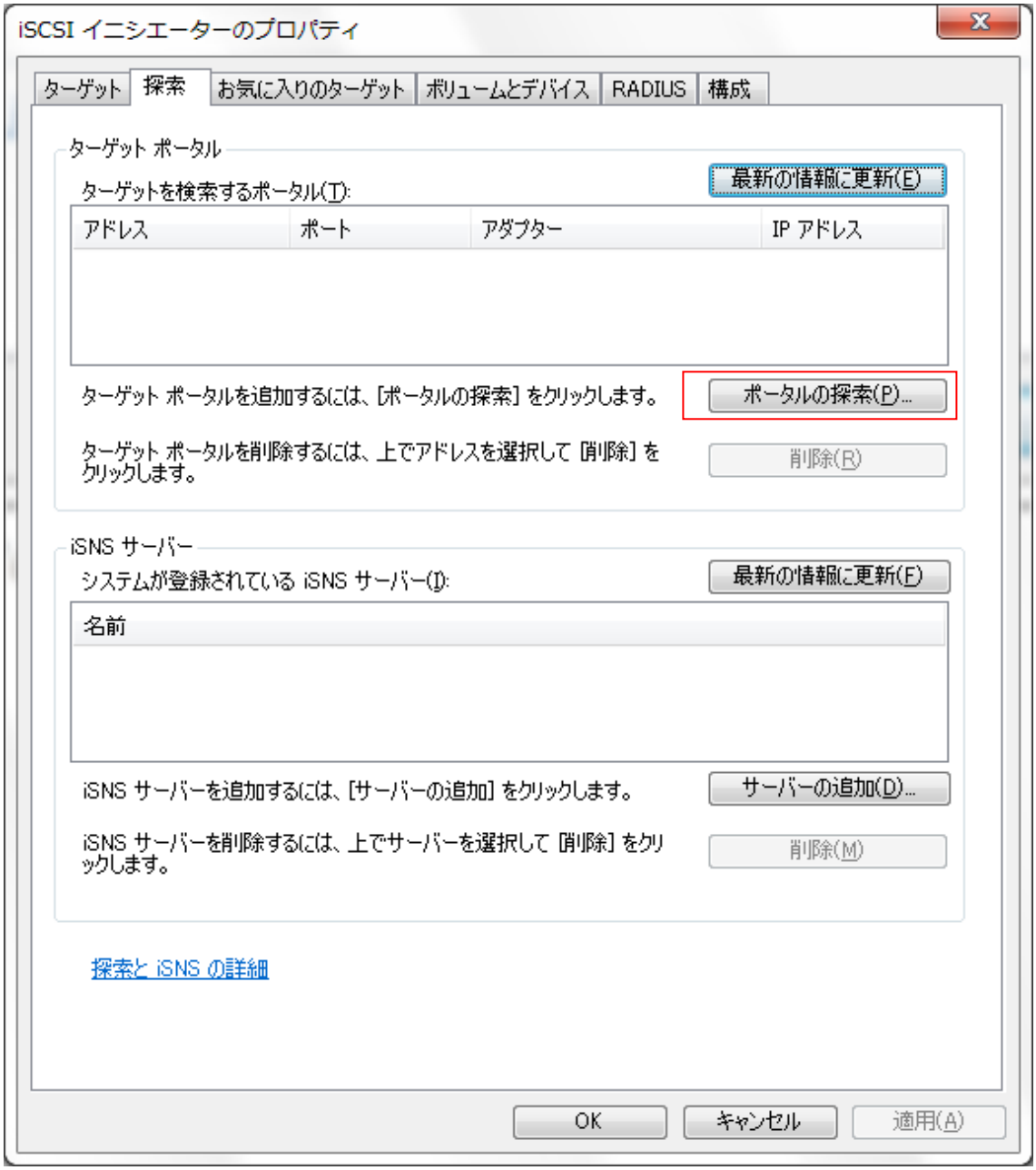
（1） iSCSI target の探索

start → コントロールパネル → システムとセキュリティ → 管理ツール → iSCSI イニシエータ

「SCSI イニシエータのロパティ」ダイアログがでます。

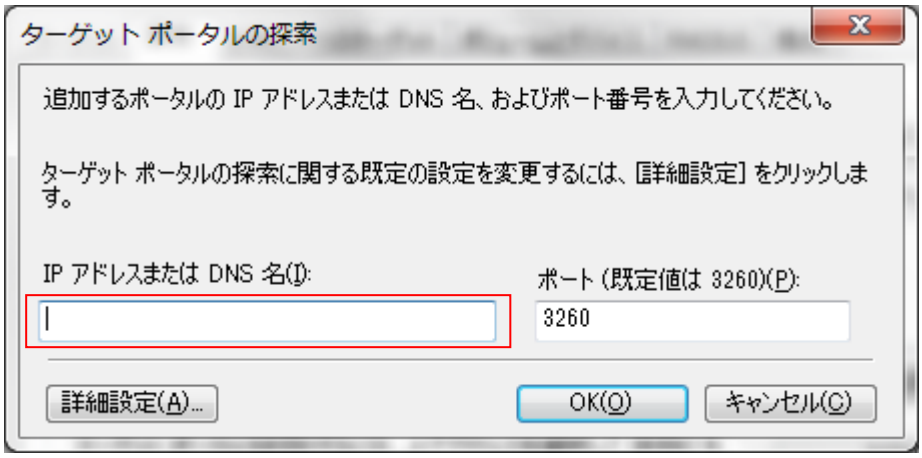


探索タグの画面を出します。

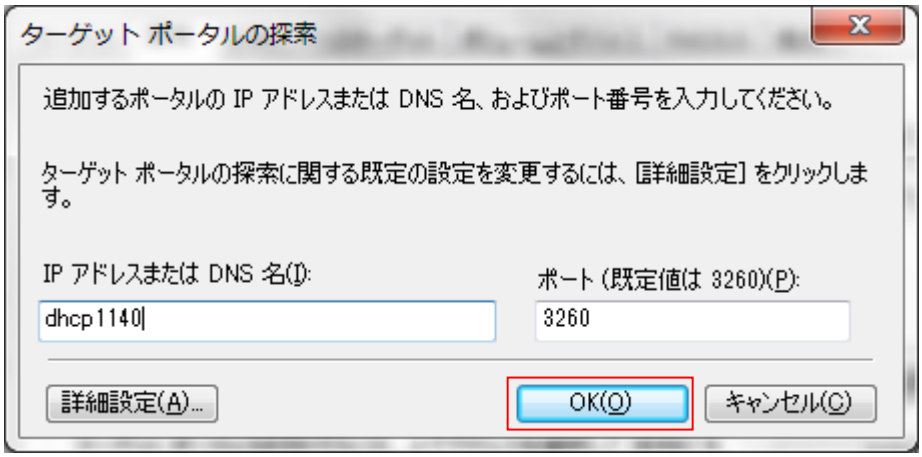


「ポータルの探索」ボタンを押します。

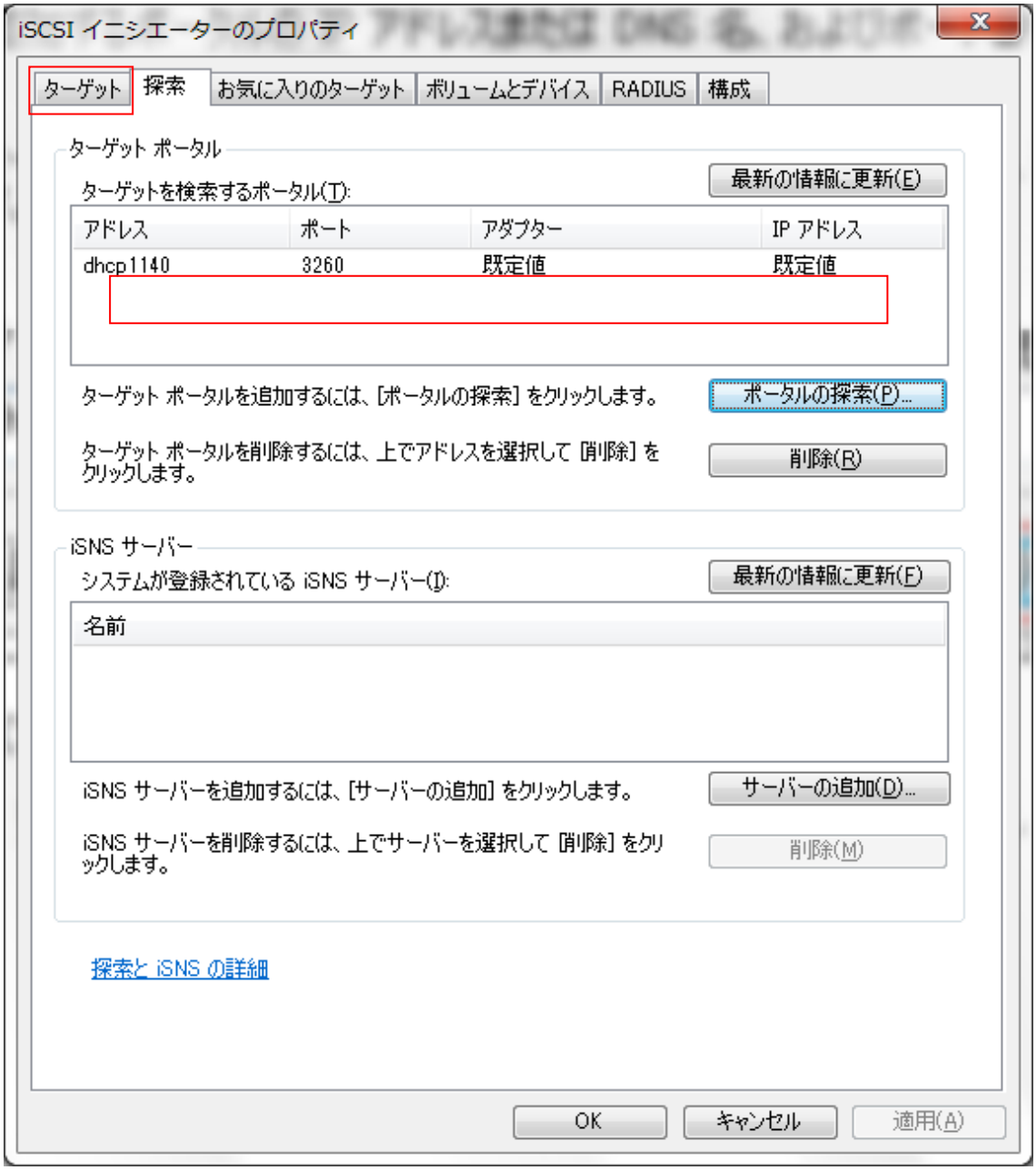
「ターゲット　ポータルの探索」ダイアログが表示されます。



iSCSI target の「IP アドレス又は DNS 名」の入力フィールを入力して「OK」ボタンを押します。

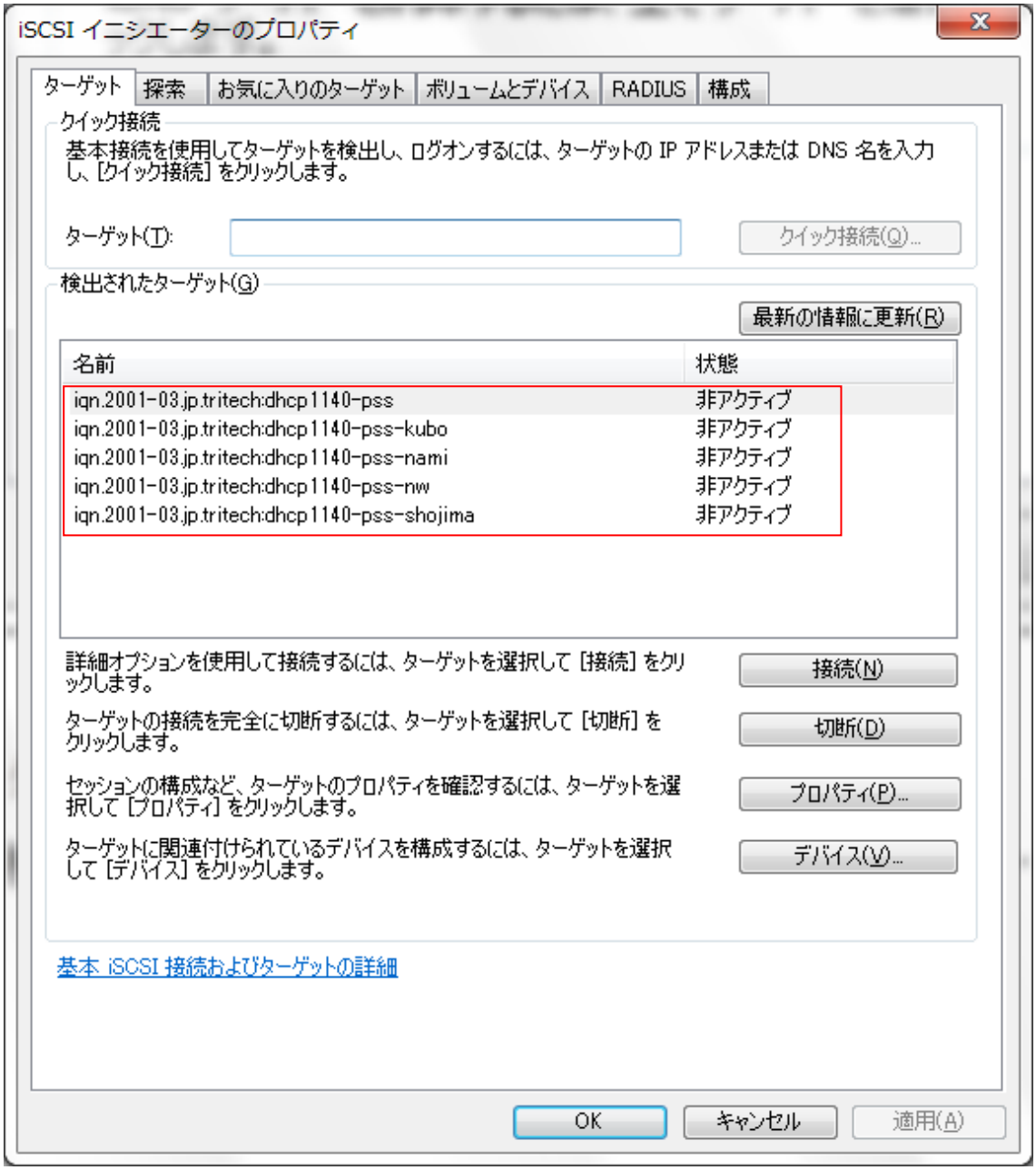


「SCSI イニシエータのロパティ」ダイアログの「ターゲットを探索するポータル」リストボックスに入力データが表示されます



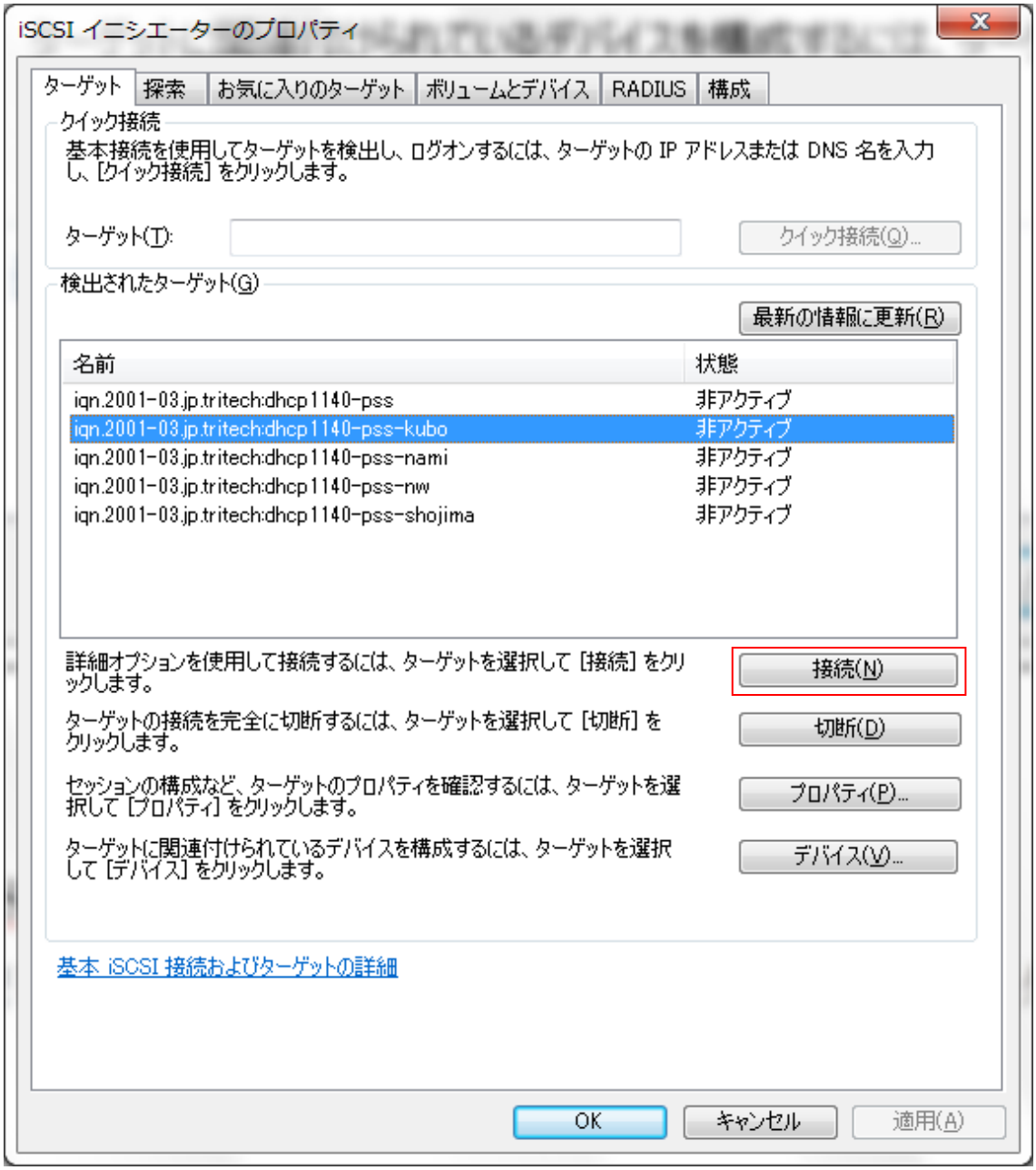
「ターゲット」タグの画面を出します。

「検出されたターゲット」リストボックスにターゲット名リストが表示されます。

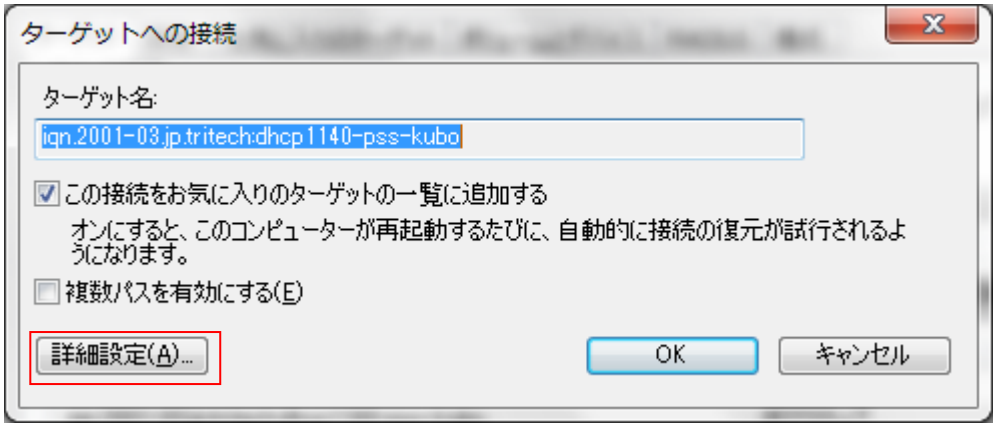


（２） iSCSI target への接続

接続したい「ターゲット」を選択して「接続」ボタンを押します。

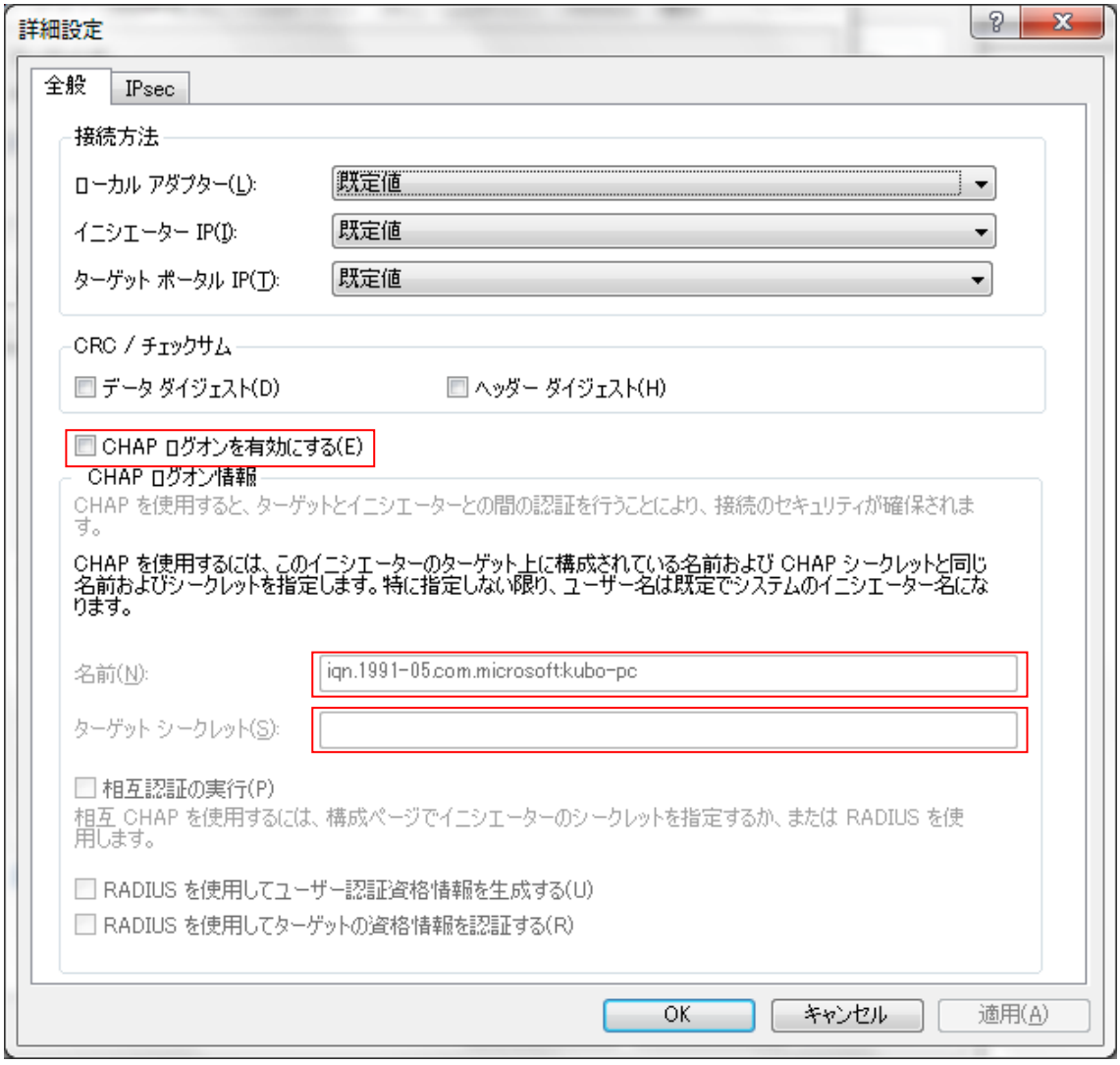


「ターゲットへの接続」ダイアログが出ます。



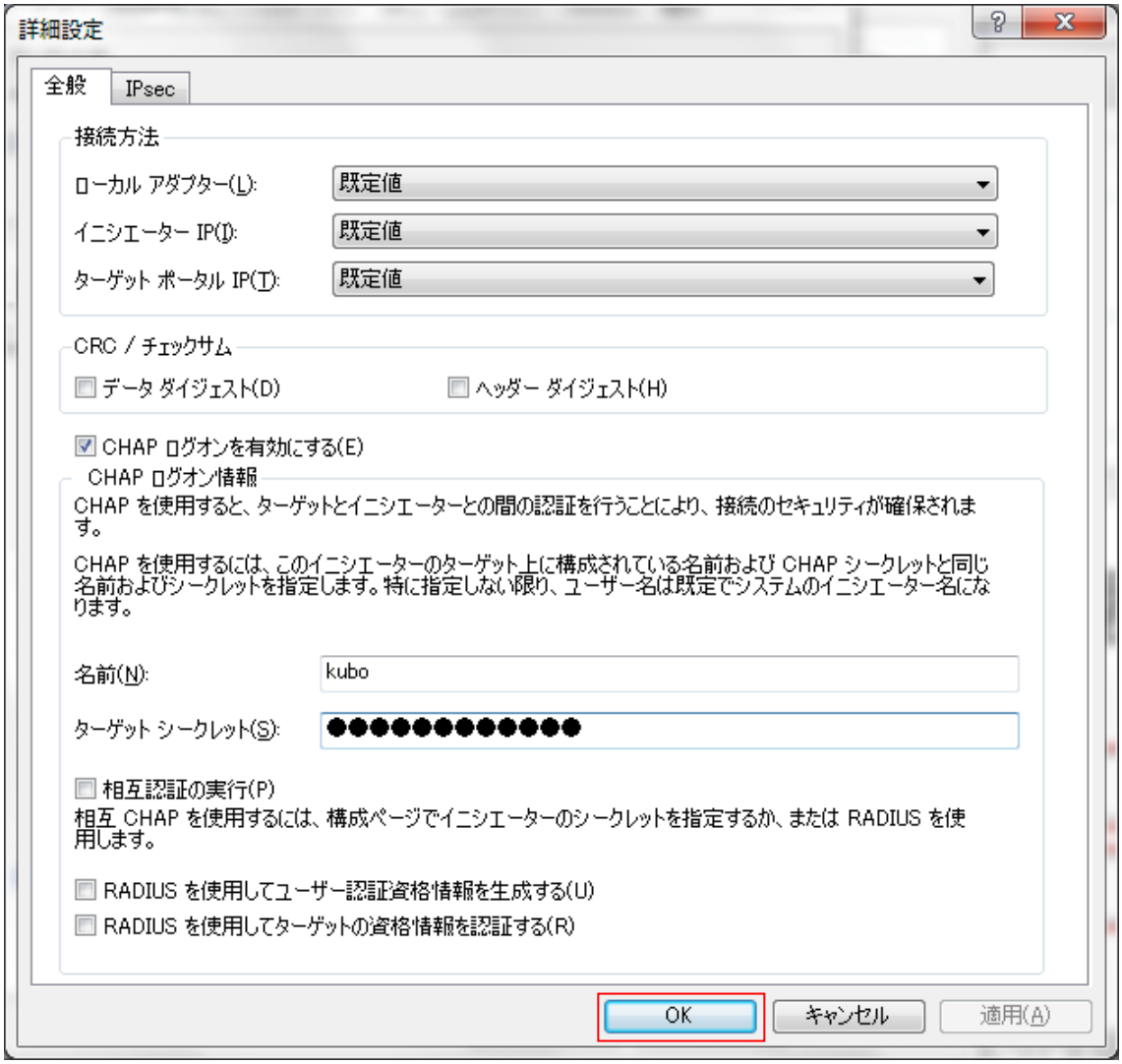
「詳細設定」ボタンを押します。

「詳細設定」ダイアログが出ます。



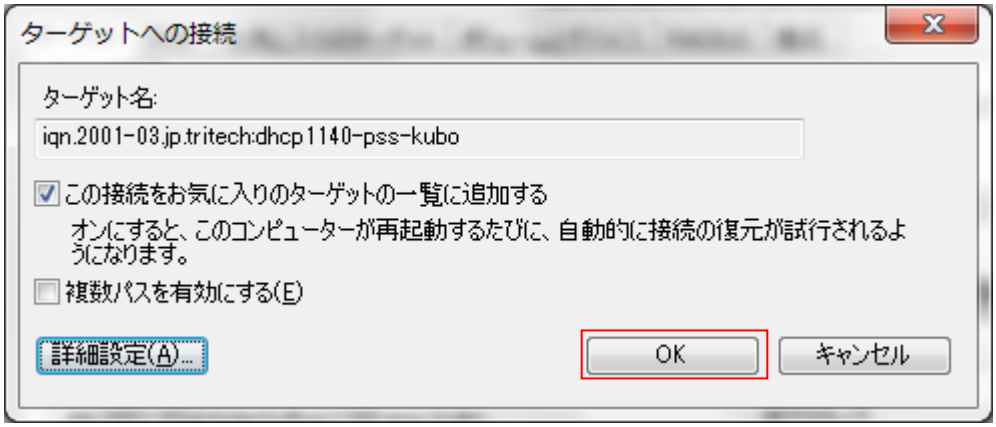
「CHAP　ログオンを有効にする」チェックボックスをチェックし、名前とターゲット　シークレットの入力フィールドに 1 ISCSI target で指定したアカウント

のデータを入力して「OK」ボタンを押します。

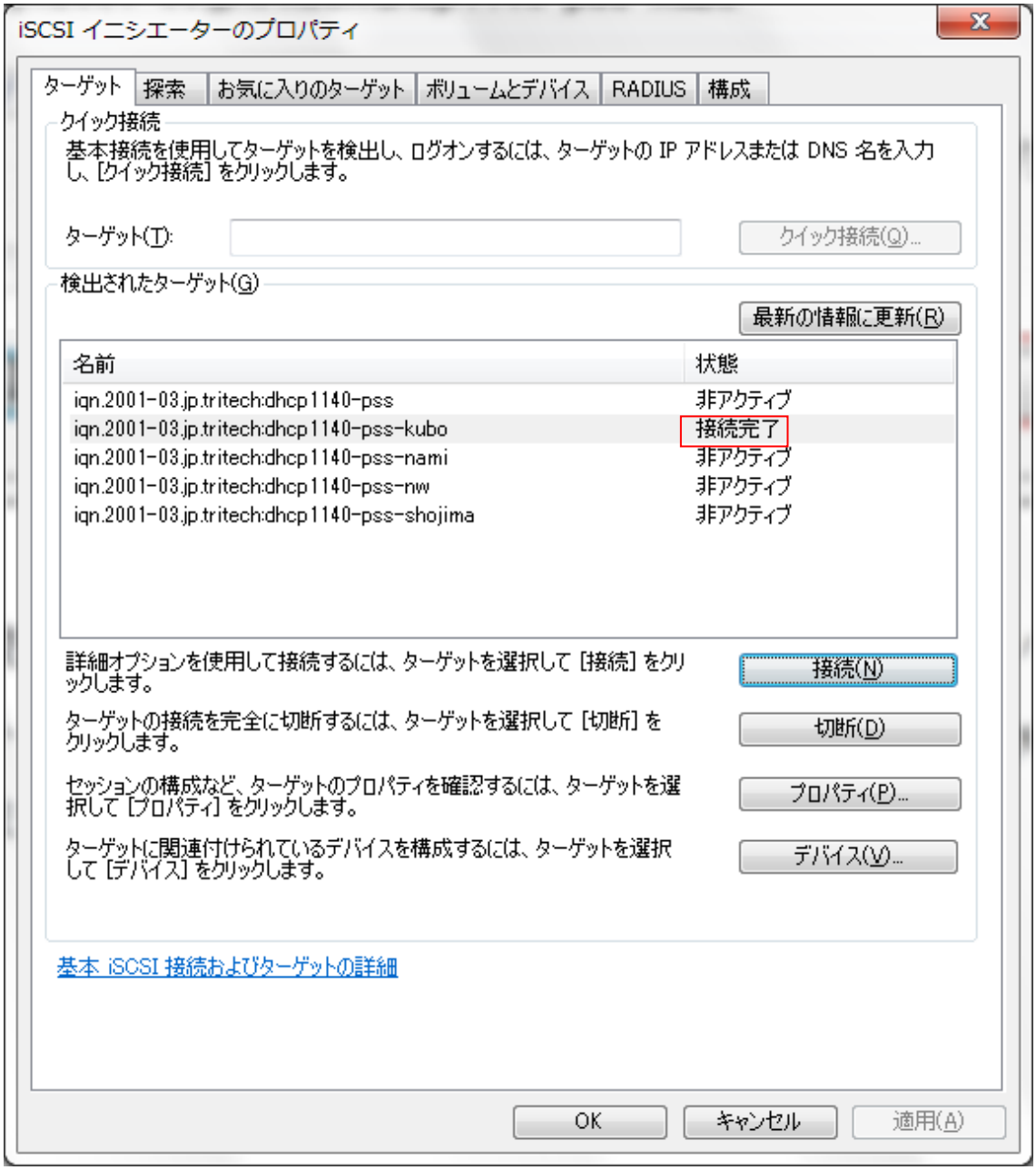


※ Windows の場合「ターゲットシークレット」は、12 文字～16 文字と文字数の制限がありますので、iSCSI target での設定に注意してください。

「OK」ボタンを押すと、「ターゲットへの接続」ダイアログ画面に戻りますので、「OK」ボタンを押します。



接続が成功すると、「iSCSI イニシエータのプロパティ」ダイアログの「ターゲット」タグ画面に戻り、対象にした「iSCSI target」の状態が「接続完了」になります。

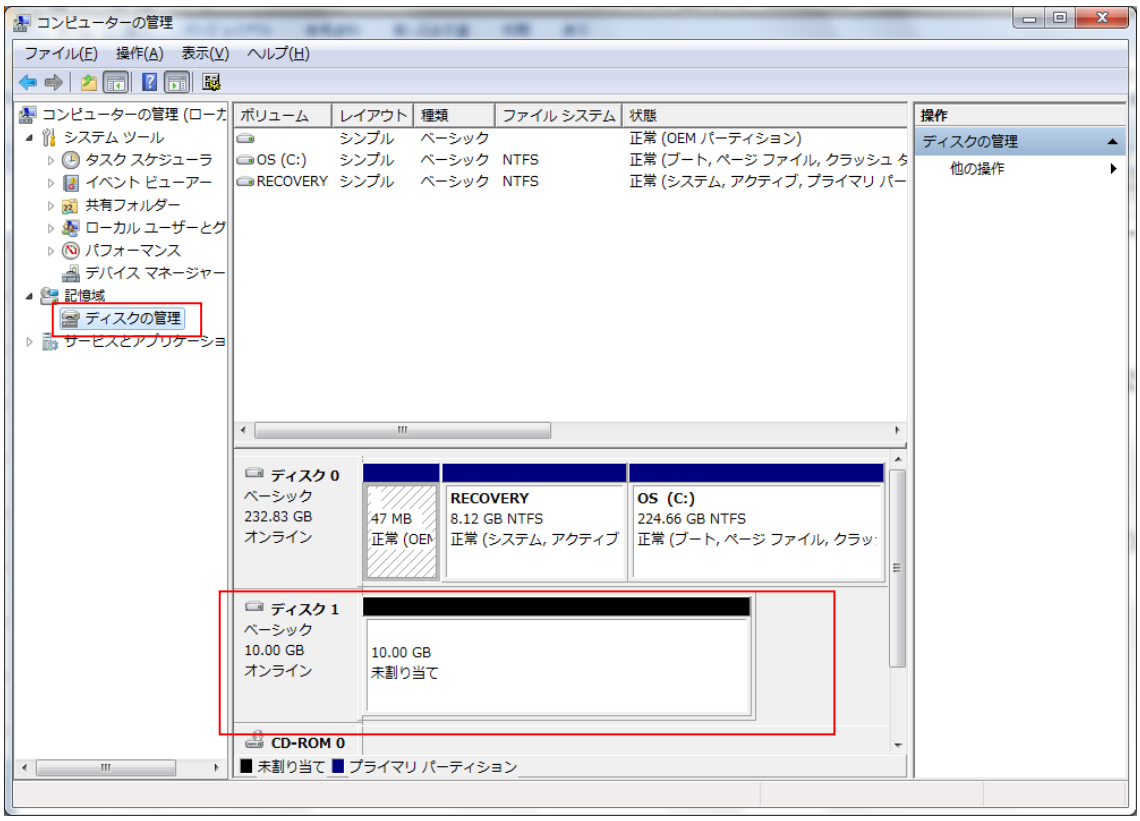


（3） フォーマット

接続ができたら、見えるようになりますが、最初は未フォーマットなので、

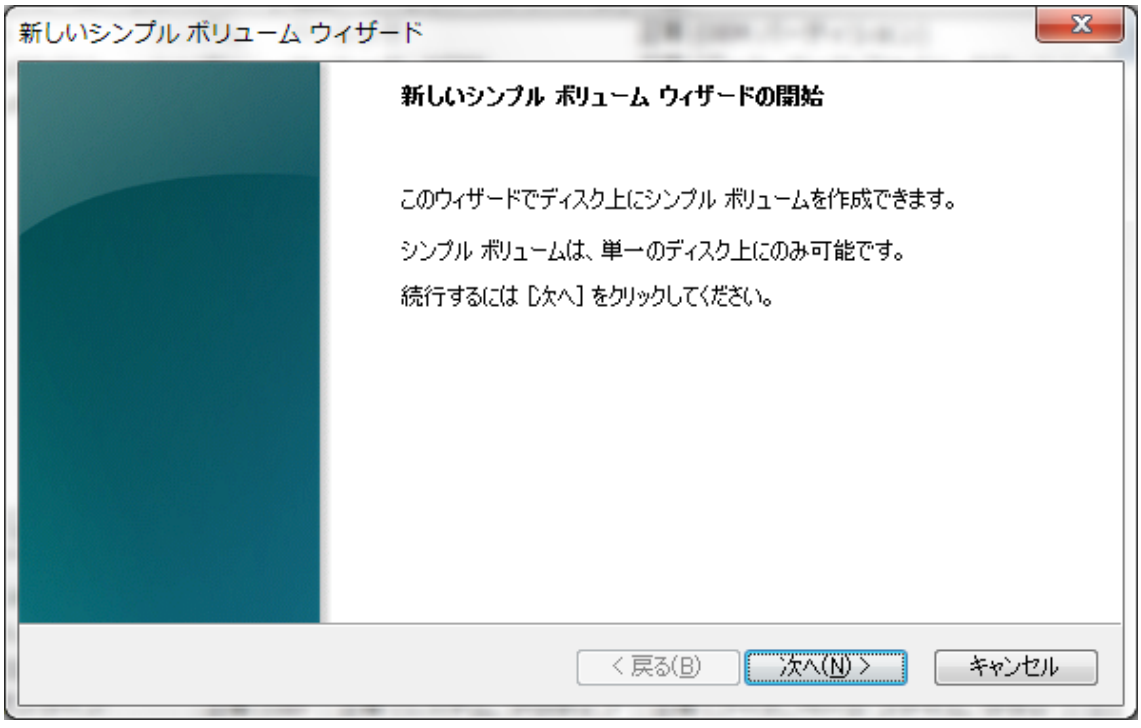
スタート　→　コントロールパネル　→　システムとセキュリティ　→　管理ツール　→　コンピュータの管理

「コンピュータの管理」ダイアログ画面の　記憶域　→　ディスクの管理

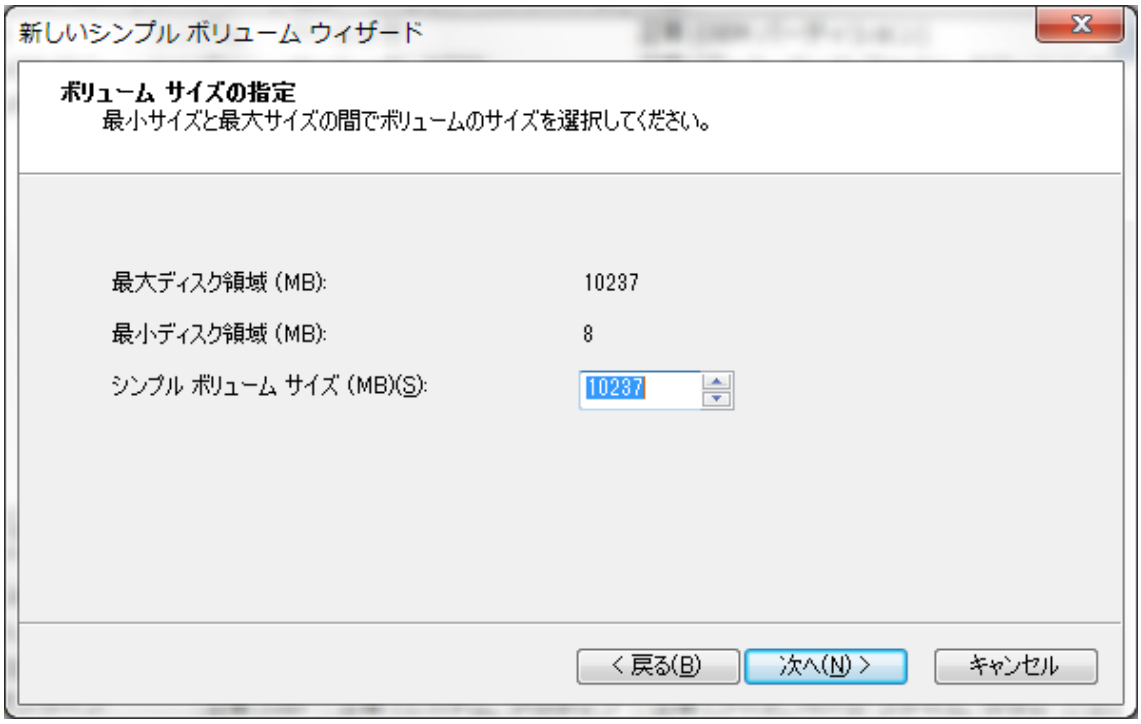


未フォーマットディスクを指定して、「新しいシンプルボリューム」を選択する。

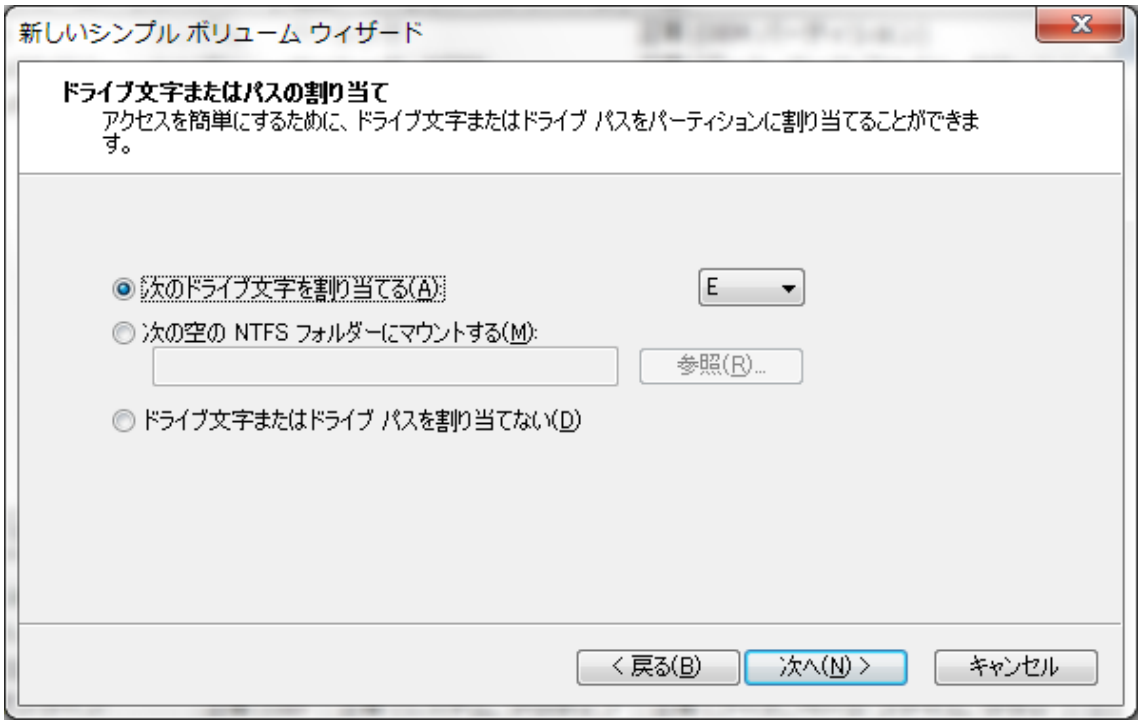
※ 「操作」→ 「タスク」→ 「新しいシンプルボリューム」か、マウス右クリック→ 「新しいシンプルボリューム」で ウイザードが起動



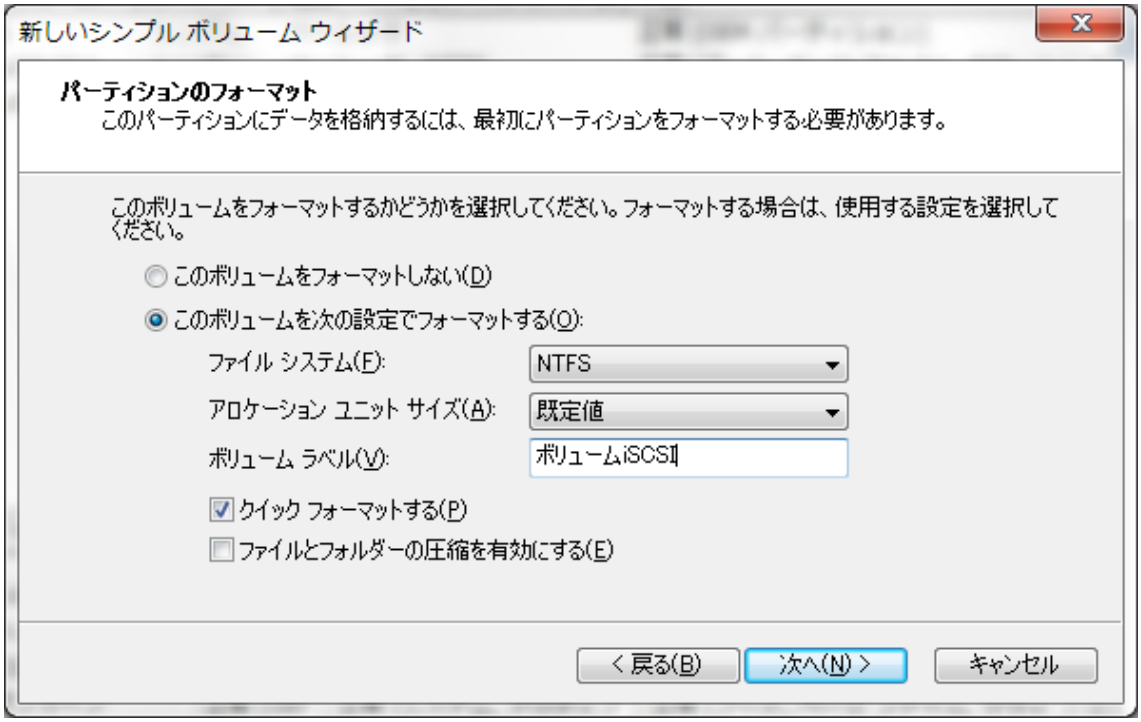
「次へ」ボタンを押します。



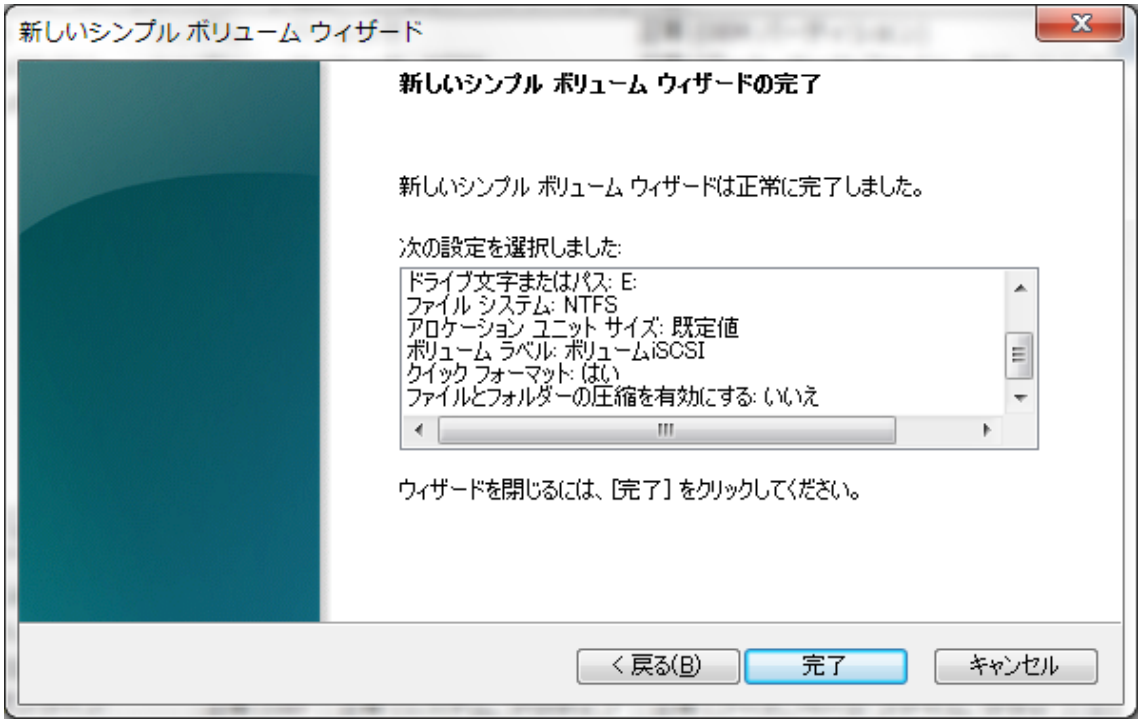
サイズを指定して、パーティション分けも可能ですが、例としてフルサイズを選択して「次へ」ボタンを押します。



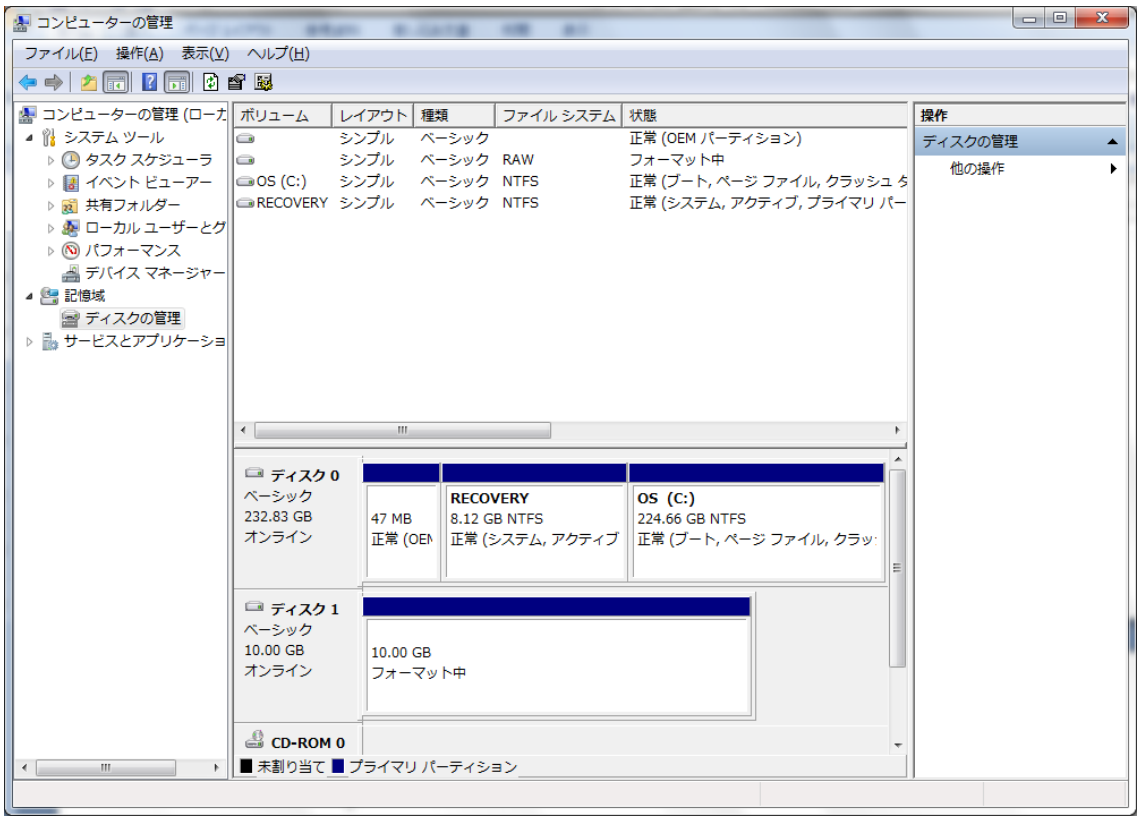
ドライブ追加する場合は、そのまま「次へ」ボタンを押します。



「ファイルシステム」「アロケーションユニットサイズ（ブロックサイズ）」「ボリューム名」を指定して「次へ」のボタンを押します。



問題がなければ「完了」を押します。



ウィザード終了して、「コンピュータの管理」の「ディスクの管理」画面で対象ディスクが「フォーマット中」と出ます。

終了したら、情報が表示される